REMARKS

Claims 1-29 presently stand rejected. Claims 13 and 22 are amended herein. Entry of these amendments and reconsideration of the pending claims are respectfully requested.

Claim Rejections – 35 U.S.C. § 112

Claim 22 stands rejected under 35 U.S.C. § 112, second paragraph. Claim 22 has been amended in accordance with the Examiner's interpretation.

Claim Rejections – 35 U.S.C. § 103

Claims 1-3, 5, 7-11, 13, 14, 16, 17, 19, 20, 22-24, 26, and 27 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Inoue et al. (US 5,437,033) in view of "Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor" (Sugarman). Claim 1 recites:

1. A method, comprising:

loading a virtual machine monitor (VMM) to support a service virtual machine (VM) and a guest VM of a computer system;

invoking a service operating system (OS) in the service VM **during the pre-boot phase of the computer system**, the service OS to allow observation of a guest OS;

invoking the guest OS in the guest VM; and

switching between the guest VM and the service VM during an OS runtime of the guest OS.

In contrast the combined references do not teach or suggest the limitations emphasized above. Sugarman does not teach or suggest invoking a service OS in the service VM during the preboot phase of the computer system. Sugarman instead teaches a hosted architecture that allows it to co-exist with a preexisting host operating system, and rely upon that operating system for device support. Figure 2 of Sugarman shows a

hosted virtual machine model that splits the virtualization software between a virtual machine monitor that virtualizes the CPU, an application [that] uses a host operating system for device support, and an operating system driver for transitioning between them.

In the second paragraph of section 2 on page 3, Sugarman teaches the Vmware workstation installs like a normal application on an operating system, known as the host operating system. When run, the application portion (VMApp) uses a driver loaded into the host operating system (VMDriver) to establish the privileged virtual machine monitor component (VMM) that runs directly on the hardware. Accordingly the OS is loaded before the VMM so that the VMApp and Driver can be loaded and executed by the OS. The OS in the service VM cannot be invoked during the pre-OS phase because the specification defines the pre-boot phase as the phase from the firmware that executes between the processor reset and the first instruction of the Operating System (OS) loader. Thus, the service operating system (OS) in the service VM is not invoked **during the pre-boot phase of the computer system**, the service OS to allow observation of a guest OS.

Inoue, either singly or in motivation combination with Sugarman, fails to overcome the deficiency because Inoue is directed towards providing a method and system for continuous operation of a computer system that is capable of continuing the operation of a specific virtual machine even if a failure occurs due to a program error of the the virtual machine monitor (see, Summary of the Invention). Inoue does not teach or suggest an operating system (OS) in the service VM that is invoked **during the pre-boot phase of the computer system.**

Furthermore the office action concedes that Sugarman fails to teach or suggest switching between the guest VM and the service VM during an OS runtime of the guest OS. Inoue fails to overcome this deficiency (singly or in motivated combination) because the "Start Interpretive Execution" instruction is merely used to perform a switch from the nonguest mode to the guest mode.

As per claim 8, Sugarman fails to teach or suggest wherein the VMM is loaded during the pre-boot phase of the computer system. Sugarman merely teaches that VMMs can take complete control of the hardware (no mention is made as to when this

occurs) and that each VM can behave like a physical machine that can run is own OS. Sugarman teaches that Vmware Workstation installs like a normal application on an operating system (page 2, second full paragraph), which thus cannot be loaded during the pre-boot phase.

Claim 13 recites:

13. An article of manufacture comprising:

a machine-readable medium including a plurality of instructions which when executed perform operations comprising:

loading a virtual machine monitor (VMM) during a pre-boot phase of a computer system, the VMM to support a service virtual machine (VM) and a guest VM of the computer system;

booting a service operating system (OS) during a pre-boot phase of a computer system in the service VM, wherein the service OS to provide tools to diagnose a guest operating system of a computer system;

booting the guest OS into the guest VM to begin a guest OS runtime of the computer system; and

performing a VM switch between the guest VM and the service VM during the guest OS runtime without rebooting the computer system.

Sugerman, either singly or in motivated combination with Inoue, fails to teach booting a service operating system (OS) during a pre-boot phase of a computer system in the service VM, wherein the service OS to provide tools to diagnose a guest operating system of a computer system. As discussed above Sugarman runs the VMware application after the OS system has been booted.

Claim 22 recites:

22. A computer system, comprising:

a processor; and

at least one flash device operatively coupled to the processor, the at least one flash device including firmware instructions which when executed by the processor perform operations comprising:

loading a virtual machine monitor (VMM) on the computer system during a pre-boot phase of the computer system;

booting a service operating system (OS) in a service virtual machine (VM) during the pre-boot phase, the service OS to enable analysis of the computer system;

booting a guest OS in a guest VM of the computer system in response to a VM switch from the service OS to the guest OS; and

performing the VM switch from the guest VM to the service VM during an OS runtime of the guest OS in response to a fault of the guest OS.

Sugerman, either singly or in motivated combination with Inoue, fails to teach booting a service operating system (OS) in a service virtual machine (VM) during the pre-boot phase, the service OS to enable analysis of the computer system. As discussed above Sugarman runs the VMware application after the OS system has been booted.

At least for the foregoing reasons, the combined references do not teach or suggest each and every element of independent claims 1, 13, and 22. Therefore claims 1, 13, and 22 and all claims which depend therefrom are patentable over the combination of references.

CONCLUSION

In view of the foregoing amendments and remarks, it is believed that the applicable rejections have been overcome and all claims remaining in the application are presently in condition for allowance. Accordingly, favorable consideration and a Notice of Allowance are earnestly solicited. The Examiner is invited to telephone the undersigned representative at (206) 292-8600 if the Examiner believes that an interview might be useful for any reason.

CHARGE DEPOSIT ACCOUNT

It is not believed that extensions of time are required beyond those that may otherwise be provided for in documents accompanying this paper. However, if additional extensions of time are necessary to prevent abandonment of this application, then such extensions of time are hereby petitioned under 37 C.F.R. § 1.136(a). Any fees required therefore are hereby authorized to be charged to Deposit Account No. 02-2666. Please credit any overpayment to the same deposit account.

Respectfully submitted,
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

Date: Sept 14, 2007

Mark R. Hennings
Reg. No. 48,982
Phone: (206) 292-8600

Blakely Sokoloff Taylor & Zafman LLP
1279 Oakmead Parkway
Sunnyvale, California
94085-4040